



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (1)

مفردات المحاضرة :

مقدمة عن لغة Hypertext Preprocessor PHP

مفهوم – لغة البرمجة بي إتش بي PHP

Server Side Scripting Language

إمكانيات لغة PHP

أهداف المحاضرة :

بعد الانتهاء من عرض و شرح مفردات المحاضرة ستتكون عند الطالب فكرة واضحة

عن لغة البرمجة PHP و الهدف من تعلمها .

□ مقدمة تاريخية عن لغة Hypertext PreProcessor PHP

تأسيس وتاريخ PHP

- أنشئت أصلاً من قبل راسموس ليردورف Rasmus Lerdorf في عام 1994، وأصدرت بعد ذلك بعام. بعد عدّة أعوام، أعاد اثنان من المبرمجين الجامعيين، وهما "زئيف سوراسكي" و"أندي غوتمانز"، كتابة قاعدة النسخة الأصلية، وأطلقا الإصدار الذي سُمي حينها بـ "PHP3".

دلالة مصطلح PHP

- التفسير الأكثر شعبيةً لما تمثّله PHP هو ((PHP: Hypertext PreProcessor))، ولكن هذا من شأنه أن يجعل المصطلح المعبر عنها بالشكل: ((HPP))، وهذا أقرب لأن يكون اختصاراً عكسياً!
- عموماً؛ فالتفسير البديل هو أنّ الأحرف الأولى تأتي من الإصدار الأقدم لبرنامج اللّغة، والذي كان يُسمّى - بالنظر إلى الغرض منه - بأدوات "الصفحة الشخصية الرئيسية":
(Personal HomePage).

مفهوم – لغة البرمجة بي إتش بي PHP

- هي لغة برمجة نصية كانت في الأصل مجموعة من التطبيقات التي كتبت باستخدام لغة Perl، وهي اليوم تُعرف بأنها لغة برمجة نصية من جانب الخادم .
- لغة المعالج السابق للنص التشعبي و بالإنجليزية Hypertext Pre Processor أو كما تعرف بـ PHP هي لغة برمجة تم صناعتها أساسا من أجل استخدامها لتطوير و برمجة مواقع الويب.
- يمكن استخدامها أيضا لأغراضٍ عامّةٍ أكثر شمولاً، كإنتاج برامجٍ قائمةٍ بذاتها وليس لها علاقةٌ بالويب فقط .
- تعتبر لغة PHP من اقوى اللغات الحالية واسرعها , وتعتبر المفضلة عند كثير من المبرمجين لسهولةتها وبساطة التعامل معها.

Server Side Scripting Language

س : ما معنى "لغة برمجة من جانب الخادم"؟!؟

إن التطبيق الأساسي والأكثر إبداعاً لـ PHP يكون عندما يتم استخدامها من ناحية الخادم Server Side؛ إذ تمّ تصميمها في واقع الأمر لهذا الغرض، من أجل بناء مواقع وتطبيقات ويب تفاعلية، بطريقة آمنة، غير مكلفة، ويمكن الاعتماد عليها.

انتشار وشعبية PHP

عندما نتحدث عن الإنترنت اليوم، فإننا أمام مصطلح من الصعب جدًا قياسه، وإحصاء تنوعه، وموارده. لكن PHP لها اليد العليا أمام كل هذا!

إنها قوية بما يكفي لتكون في قلب أكبر نظام تدوين على الويب
WordPress

إنها عميقة بما يكفي لتشغيل أكبر شبكة اجتماعية في العالم
Facebook

إنها واسعة ومتشعبة بما يكفي لتكوين أكبر موسوعة معارف إلكترونية في العالم
Wikipedia

لغة برمجة PHP هي لغة عامة الأغراض

عندما يتعلق الأمر بالغرض من لغات البرمجة, يوجد هنالك نوعان رئيسيان:

• اللغات الخاصة بالنطاق. Domain-Specific Languages.

• لغات الأغراض العامة. General-Purpose Languages.

يتم استخدام اللغات الخاصة بالنطاق في مجالات معينة. على سبيل المثال, SQL هي لغة خاصة

بالنطاق. يتم استخدامها بشكل أساسي للاستعلام عن البيانات من قواعد البيانات, ولا يمكن استخدام

SQL لأغراض أخرى.

ومن ناحية أخرى, تعتبر لغة البرمجة PHP هي لغة عامة الغرض لأن PHP يمكنها تطوير

تطبيقات مختلفة.

PHP هي لغة مشتركة بين الأنظمة الأساسية

- يمكن تشغيل PHP على جميع أنظمة التشغيل الرئيسية, وهي تعتبر لغة مشتركة بين الأنظمة الأساسية, Cross-Platform Language, بما في ذلك أنظمة التشغيل Linux وWindows وmacOS.
- يمكن استخدام PHP مع جميع خوادم الويب الرائدة مثل Nginx وOpenBSD وApache. كما تدعم بعض البيئات السحابية أيضاً لغة PHP مثل Microsoft Azure وAmazon AWS. PHP.
- لا يقتصر الأمر على معالجة HTML فقط. تحتوي لغة PHP على دعم مدمج لإنشاء صور PDF وGIF وJPEG. PNG. تتمثل إحدى الميزات البارزة للغة PHP في أنها تدعم العديد من قواعد البيانات. بما في ذلك MySQL وPostgreSQL وMS SQL وdb2 وOracle Database وMongoDB.

إمكانيات لغة PHP





**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٢)

مفردات المحاضرة :

- Basic PHP Syntax ، قواعد PHP الاساسية
- Comments in PHP التعليقات
- PHP echo and print Statements دوال الاخراج
- PHP Variable المتغيرات
- طرق تضمين شفرات PHP في HTML

Basic PHP Syntax □ ، قواعد PHP الأساسية

➤ A PHP script starts with `<?php` and ends with `?>` :

```
<?php
// PHP code goes here
?>
```

➤ PHP Case Sensitivity

```
<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
Echo "Hello World!<br>";
?>
```

في PHP ، الكلمات الرئيسية
(e.g. if, else, while, echo, etc.),
classes, functions, and user-
defined functions

المعرفة من قبل المستخدم ليست حساسة لحالة
الأحرف.

Note: However; all variable names
are case-sensitive!

التعليقات Comments in PHP

➤ التعليق في PHP هو سطر لا يتم تنفيذه كجزء من البرنامج. الغرض منه هو أن يعطي فكرة عن الكود المستخدم .

```
<?php
// This is a single-line comment

# This is also a single-line comment
?>
```

نستخدم // او # لتعليق سطر برمجي واحد

```
<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>
```

لتعليق عدة اسطر نستخدم /* في بداية السطر الاول و */ في نهاية السطر الاخير

□ PHP echo and print Statements دوال الاخراج

في php يوجد دالتين مهمة لإخراج البيانات للمتصفح وهي :

• الدالة echo

• الدالة print

يوجد إختلافات بسيطة بين دالة **echo** و **print** عند الإستخدام ممكن تلخيصها بالتالي:

• الدالة **echo** تستطيع إخراج و طباعة نص أو أكثر من نص مفصولين بفاصلة .

• الدالة **print** تستطيع إخراج و طباعة نص واحد فقط وعند تنفيذ الأمر بنجاح يتم إرجاع القيمة ١ كعلامة

لنجاح تنفيذ الأمر.

• الدالة **echo** أسرع من الدالة **print** لأنها لاتقوم بإرجاع أي قيمة إنما تنفذ الأمر بطباعة النص فقط.

الدالة echo

ويمكن إستخدامها بأقواس كبقية الدوال أو بدون language construct من أساس اللغة echo الدالة أقواس

❑ PHP echo and print Statements دوال الاخراج

➤ The PHP **echo** Statement

الدالة **echo** من أساس اللغة language construct ويمكن استخدامها بأقواس كبقية الدوال أو بدون أقواس :

➤ Display Text

```
<?php  
echo "<h2>PHP is Fun!</h2>";  
echo "Hello world!<br>";  
echo "I'm about to learn PHP!<br>";  
echo "This ", "string ", "was ", "made ", "with multiple parameters."  
?>
```



PHP is Fun!
Hello world!
I'm about to learn PHP!
This string was made with multiple parameters.

طباعة كلمة معينة من النص
محصورة بعلامتي اقتباس



```
echo "Hello escape \"sequence\" characters";
```

❑ PHP echo and print Statements دوال الاخراج

- PHP echo: printing variable value

```
<?php  
$msg="Hello php world ";  
echo "Message is: $msg";  
?>
```



Message is: Hello php world

- The PHP **print** Statement

الدالة **print** من أساس اللغة language construct ويمكن استخدامها بأقواس كبقية الدوال أو بدون أقواس :

❑ PHP echo and print Statements دوال الاخراج

➤ Display Text

```
<?php
<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>?>
```



PHP is Fun!
Hello world!
I'm about to learn PHP!

طباعة كلمة معينة من النص
محصورة بعلامتي اقتباس



```
echo "Hello escape \"sequence\" characters";
```

❑ PHP echo and print Statements دوال الاخراج

➤ PHP print: printing variable value

```
<?php
$txt1 = "Learn PHP";
$txt2 = "Qurna Technical Institute";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```

Learn PHP

Study PHP at Qurna Technical Institute

9

طباعة كلمة معينة من النص
محصورة بعلامتي اقتباس

```
print "Hello escape \"sequence\" characters";
```



➤ Variables are "containers" for storing information.

➤ Creating (Declaring) PHP Variables

قواعد تسمية المتغيرات في PHP:

- يبدأ المتغير بعلامة الدولار \$ متبوعا باسم المتغير.
- يجب ان يبدأ المتغير بحرف او ب خط تحتي (_) علامة الاندرسكور.
- لايمكن ان يبدأ اسم المتغير برقم ابدا (قم بوضع علامة _ قبل الرقم ليتم قبوله) .
- يجب ان يتكون الأسم من حروف ابجدية وارقام او بعلامة (_) .
- اسماء المتغيرات حساسه بالكتابه مثلا \$ y ليس مثل \$ Y لكل متغير قيمته الخاصة .
- يُستخدم عامل الاسناد (=) لتعيين قيمة إلى متغير.
- لا يمكن أن يحتوي اسم المتغير على مسافات.

PHP Variables □

أمثلة متنوعة ➤

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

اسناد خيط رمزي الى متغير

اسناد قيمة من النوع الصحيح الى متغير

اسناد قيمة من النوع الحقيقي الى متغير

```
<?php
$txt = " Qurna Technical Institute ";
echo "I love $txt!";
?>
```

✓ طباعة نص و متغير

```
<?php
$txt = " Qurna Technical Institute ";
echo "I love " . $txt . "!";
?>
```

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

✓ في PHP لا داعي للإعلان عن أنواع بيانات المتغيرات .

ملحق محاضرة (2)

توجد ثلاث طرق لتضمين شفرات PHP في HTML

1. Without any HTML markups

```
<?php
  echo "<h1> welcome </h1>" ;
?>
```

2. Embedding HTML markups in PHP code:

```
<?php
  echo "<html>";
  echo "<h1> welcome </h1>" ;
  echo "</html>" ;
?>
```

3. Embedding PHP code in HTML:

```
<!DOCTYPE html>
<html>
<body>
<?php
$txt = "Hello world!";
echo $txt;
</body>
</html>
```



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٣)

- PHP Variables
- PHP Data Types
 - PHP String
 - PHP Number: integer
 - PHP Number: float
- PHP Math

□ PHP \$ and \$\$ Variables

```
<?php
$x = "Qurna" ;
$$x = 2022;
echo $x."<br/>" ;
echo $$x."<br/>" ;
echo $Qurna;
?>
```

```
Qurna
2022
2022
```

- `$var`
هو متغير عادي باسم `var` يخزن أي قيمة بداخله مثل سلسلة خيط رمزي او عدد صحيح ، أو عدد عشري ، والخ ...

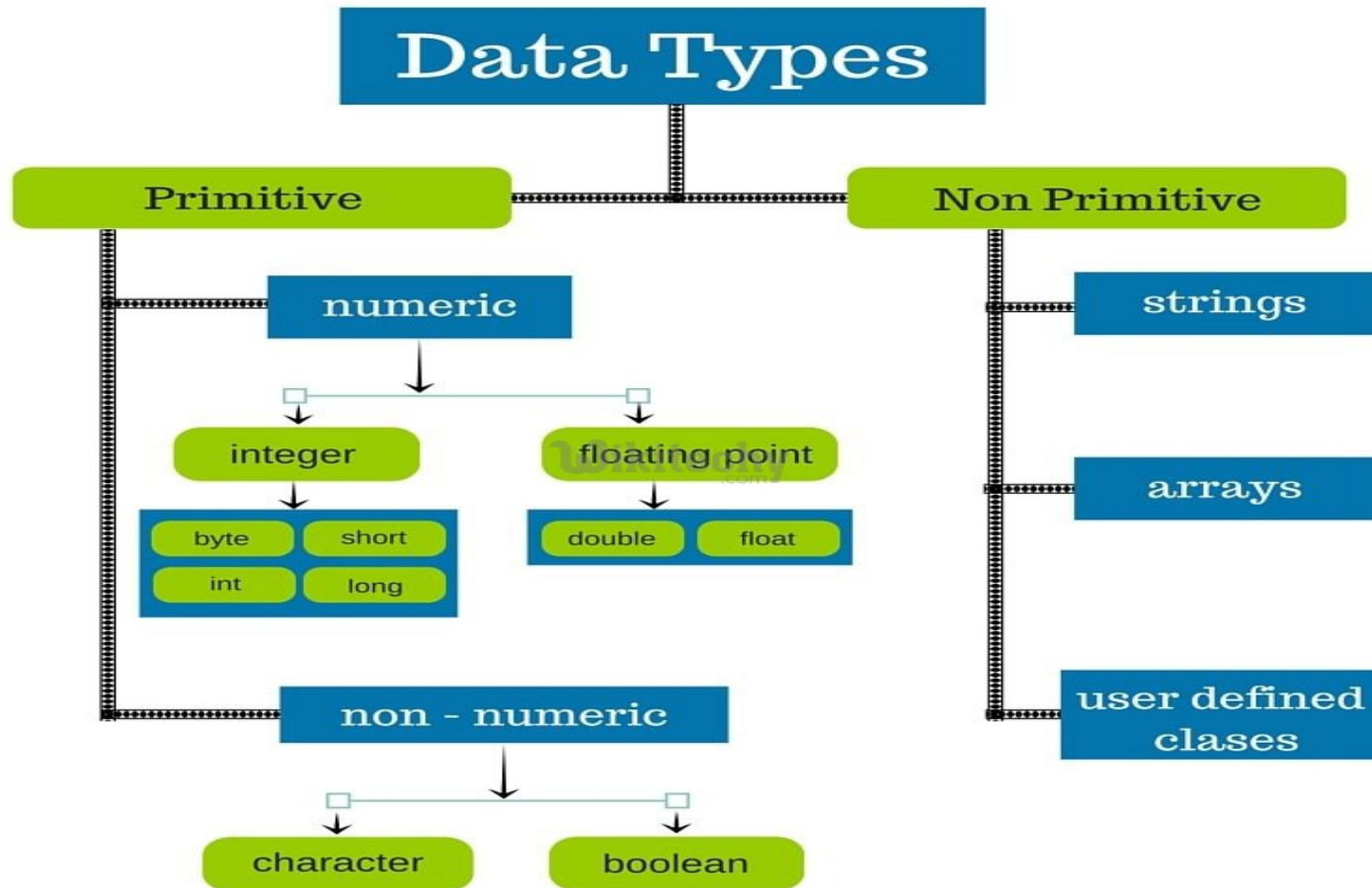
- `var $$`
هو متغير مرجعي يخزن قيمة المتغير `$` بداخله.

```
<?php
$x="Iraq";
$$x="Baghdad";
echo "Capital of $x is " . $$x ;
?>
```

```
Capital of Iraq is Baghdad
```


☐ PHP Data Types

- Variables can store data of different types, and different data types can do different things.



❑ PHP String

الخيطة الرمزي او السلسلة عبارة عن سلسلة من الأحرف ، مثل " Hello world " تكون السلسلة داخل علامات الاقتباس. يمكنك استخدام علامات الاقتباس المفردة أو المزدوجة :

```
<?php
$x = 'Computer Systems ' ;
$y = "Techniques";
//$x .= $y;
echo $x . "<br>";
echo $y;
?>
```



❑ PHP String Functions

➤ هذه الدالة تعيد طول الخيط الرمزي **strlen()**

```
<?php
echo strlen("Computer Systems"); // 16
?
```

❑ PHP String Functions

- **str_word_count()** هذه الدالة تعيد عدد الكلمات في الخيط الرمزي

```
echo str_word_count("Computer Systems"); // outputs 2
```

- **strrev()** هذه الدالة تعكس الكلمة في الخيط الرمزي

```
echo strrev("Computer Systems"); // smetsyS retupmoC
```

- **strpos()**

تبحث هذه الدالة عن نص معين داخل سلسلة. إذا تم العثور على النص المطلوب تقوم الدالة بإرجاع موقع أول رمز للنص المطلوب إذا لم يتم العثور على تطابق ، فسيتم إرجاع **FALSE**.

```
echo strpos("Computer Systems Techniques ", "Techniques"); // 17
```

❑ PHP String Functions

➤ **str_replace()** هذه الدالة تستبدل رموز او نص معين بنص اخر في الخيط الرمزي

```
echo str_replace("Systems", "Science", "Computer Systems");  
//Computer Science
```

```
<?php  
echo ucfirst("hello world!");  
echo "<br>";  
echo lcfirst("Hello world!");  
echo "<br>";  
echo strtoupper("Hello WORLD!");  
echo "<br>";  
echo strtolower("Hello WORLD.");  
?>
```

حدد عمل الدوال التالية ، و ماهي مخرجات كل دالة ؟

❑ PHP Number: integer

```
<?php
$x = 1267;
echo($x);
?>
```



قواعد للأعداد الصحيحة:

- يجب أن يحتوي العدد الصحيح على رقم واحد على الأقل.
- يجب ألا يحتوي العدد الصحيح على فاصلة عشرية.
- يمكن أن يكون العدد الصحيح موجباً أو سالباً.
- يمكن تحديد الأعداد الصحيحة في:
 - decimal (base 10), hexadecimal (base 16), octal (base 8), or binary (base 2) notation

PHP has the following functions to check if the type of a variable is integer:

```
is_int()
is_integer()
is_long()
```

```
$x = 5985;
var_dump(is_int($x));
```

```
$x = 59.85;
var_dump(is_int($x));
```



```
bool(true)
bool(false)
```

❑ PHP Number: float

رقم يحتوي على فاصلة عشرية :

```
<?php  
$x = 12.67;  
echo($x);  
?>
```

PHP has the following functions to check if the type of a variable is float:

```
is_float()  
is_double()
```

```
$x = 10.365;  
var_dump(is_float($x));
```

```
bool(true)  
bool(false)
```

❑ PHP Numerical Strings

يمكن استخدام الدالة `is_numeric()` لمعرفة ما إذا كان المتغير رقميًا. ترجع الدالة `true` إذا كان المتغير عبارة عن رقم أو سلسلة رقمية ، بينما ترجع الدالة `false` إذا كان غير ذلك.

```
$x = 5985;  
var_dump(is_numeric($x));  
  
$x = "5985";  
var_dump(is_numeric($x));  
  
$x = "59.85" + 100;  
var_dump(is_numeric($x));  
  
$x = "Hello";  
var_dump(is_numeric($x));
```

```
bool(true)  
bool(true)  
bool(true)  
bool(false)
```

ما هو عمل دالة `int_cast()` ؟



```
<?php  
$x = 23465.768;  
$int_cast = (int)$x;  
echo $int_cast;  
echo "<br>";  
$x = "23465.768";  
$int_cast = (int)$x;  
echo $int_cast;  
?>
```

☐ PHP Math

PHP has a set of math functions that allows you to perform mathematical tasks on numbers :

Math function	Usage	Example	Return
pi()	returns the value of PI	pi()	3.141592653589
min()	finds the lowest value in a list of arguments	min(0, 150, 20, -8, -200)	8 -200
max()	finds the highest value in a list of arguments	max(0, 150, 20, -8, -200)	150
abs()	returns the absolute value of a number	abs(-1)	1
sqrt()	returns the square root of a number	sqrt(9)	3
round()	rounds a number to its nearest integer	round(9.3)	9
rand()	generates a random number	rand(), rand(10, 100)	1381167960, 97

□ PHP Math

امثلة : -

```
<?php
echo(min(0, 150, 30, 20, -8, -200)); // returns -200
echo(max(0, 150, 30, 20, -8, -200)); // returns 150
?>
```

```
echo(abs(-6.7)); // returns 6.7
echo(sqrt(64)); // returns 8
echo(round(0.60)); // returns 1
echo(round(0.49)); // returns 0
```

```
echo decoct("10") . "<br>";
echo decbin("3") . "<br>";
echo dechex("1587") . "<br>";
echo intdiv(8, 4) . "<br>";
```

حدد عمل الدوال التالية ، و ماهي مخرجات كل دالة ؟



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٤)

PHP Operators

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators

❑ PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional assignment operators

☐ PHP Operators

☐ Arithmetic Operators

Operator	Name	Example	Explanation
+	Addition	$\$a + \b	Sum of operands
-	Subtraction	$\$a - \b	Difference of operands
*	Multiplication	$\$a * \b	Product of operands
/	Division	$\$a / \b	Quotient of operands
%	Modulus	$\$a \% \b	Remainder of operands
**	Exponentiation	$\$a ** \b	$\$a$ raised to the power $\$b$

❑ PHP Operators

❑ Assignment Operators

Operator	Name	Example	Explanation
=	Assign	$\$a = \b	The value of right operand is assigned to the left operand.
+=	Add then Assign	$\$a += \b	Addition same as $\$a = \$a + \$b$
-=	Subtract then Assign	$\$a -= \b	Subtraction same as $\$a = \$a - \$b$
*=	Multiply then Assign	$\$a *= \b	Multiplication same as $\$a = \$a * \$b$
/=	Divide then Assign (quotient)	$\$a /= \b	Find quotient same as $\$a = \$a / \$b$
%=	Divide then Assign (remainder)	$\$a \% = \b	Find remainder same as $\$a = \$a \% \$b$

❑ PHP Operators

➤ Arithmetic Operators

```
<?php
$a = 8
$b = 4;
echo $a + $b . "<br>";
echo $a - $b . "<br>";
echo $a * $b . "<br>";
echo $a / $b . "<br>";
echo $a % $b . "<br>";
echo $a ** $b . "<br>";
?>
```

➤ Assignment Operators

```
<?php
$a = 2 ;
$b = 4 ;
echo $a = $b . "<br>";
echo $a += $b . "<br>";
echo $a -= $b . "<br>";
echo $a *= $b . "<br>";
echo $a /= $b . "<br>";
echo $a%=$b . "<br>";
?>
```

❑ PHP Operators

❑ Increment / Decrement operators .

Operator	Name	Example	Explanation
++	Increment	++\$a	Increment the value of \$a by one, then return \$a
		\$a++	Return \$a, then increment the value of \$a by one
--	decrement	--\$a	Decrement the value of \$a by one, then return \$a
		\$a--	Return \$a, then decrement the value of \$a by on

□ PHP Operators

□ Increment / Decrement operators .

```
<?php
$x = 4 ;
echo $x ."<br>";
echo ++$x ."<br>";
echo ++$x ."<br>";
echo $x++ ."<br>";
echo $x++ ."<br>";
echo $x++ ."<br>";
echo $x ."<br>";
?>
```

4
5
6
6
7
8
9

```
<?php
$x = 8 ;
echo $x ."<br>";
echo --$x ."<br>";
echo $x-- ."<br>";
echo $x-- ."<br>";
echo --$x ."<br>";
echo $x++ ."<br>";
echo $x ."<br>";
?>
```

❑ PHP Operators

❑ Comparison operators

Operator	Name	Example	Explanation
==	Equal	<code>\$a == \$b</code>	Return TRUE if \$a is equal to \$b
===	Identical	<code>\$a === \$b</code>	Return TRUE if \$a is equal to \$b, and they are of same data type
!==	Not identical	<code>\$a !== \$b</code>	Return TRUE if \$a is not equal to \$b, and they are not of same data type
!=	Not equal	<code>\$a != \$b</code>	Return TRUE if \$a is not equal to \$b
<>	Not equal	<code>\$a <> \$b</code>	Return TRUE if \$a is not equal to \$b
>	Less than	<code>\$a < \$b</code>	Return TRUE if \$a is less than \$b
<	Greater than	<code>\$a > \$b</code>	Return TRUE if \$a is greater than \$b
=>	Less than or equal to	<code>\$a <= \$b</code>	Return TRUE if \$a is less than or equal \$b
=<	Greater than or equal to	<code>\$a >= \$b</code>	Return TRUE if \$a is greater than or equal \$b

❑ PHP Operators

❑ Comparison operators

```
<?php
$x = 100;
$y = "100";
$z = 10;
var_dump($x == $y);
var_dump($x === $y);
var_dump($x === $y);
var_dump($x <> $z);
var_dump($x > $y);
var_dump($z <= $y);
?>
```

```
bool(true)
bool(false)
bool(false)
bool(true)
bool(false)
bool(true)
```

```
var_dump ($x == $y);
Echo "$x == $y";
```



❑ PHP Operators

❑ Logical operators

Operator	Name	Example	Explanation
and	And	\$a and \$b	Return TRUE if both \$a and \$b are true
Or	Or	\$a or \$b	Return TRUE if either \$a or \$b is true
xor	Xor	\$a xor \$b	Return TRUE if either \$a or \$b is true but not both
!	Not	! \$a	Return TRUE if \$a is not true
&&	And	\$a && \$b	Return TRUE if either \$a and \$b are true
	Or	\$a \$b	Return TRUE if either \$a or \$b is true



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٥)

PHP Conditional Statements

- if statement
- if...else statement
- if...elseif...else statement
- PHP switch Statement

PHP Loops

- while - loops
- do...while Loop
- for Loop

❑ PHP Conditional Statements

- ❑ Conditional statements are used to perform different actions based on different conditions
- ❑ In PHP we have the following conditional statements:
 - **if statement** - executes some code if one condition is true
 - **if...else statement** - executes some code if a condition is true and another code if that condition is false
 - **if...elseif...else statement** - executes different codes for more than two conditions
 - **switch statement** - selects one of many blocks of code to be executed

□ PHP Conditional Statements

```
if (condition)
{
code to be executed if condition is true;
}
```

```
$t = 20 ;
if ($t < 20 )
{
    echo " Computer Systems Techniques!";
}
```


□ PHP Conditional Statements

```
if (condition)
{
code to be executed if condition is true;
}
else
{
code to be executed if condition is false;
}
```

```
$t = 20 ;
if ($t < 20 )
{
    echo " Computer Systems Techniques!";
}
else
{
    echo "Computer Science ";
}
```

□ PHP Conditional Statements

```
if (condition_1)
{code to be executed if condition (1) is true;
}
else if (condition_2)
{code to be executed if condition (2) is true;
}
else
{code to be executed if all conditions are false;
}
```

```
$t = 20 ;
```

```
if ($t < 20 ) {
    echo " Computer Systems Techniques!";
} elseif ($t > 20 ) {
    echo "Qurna Technical Institute ";
} else {
    echo "Computer Science ";
}
```

❑ PHP switch Statement

Use the `switch` statement to **select one of many blocks of code to be executed**.

```
switch (n) {  
case label1:  
code to be executed if n=label1;  
break;  
case label2:  
code to be executed if n=label2;  
break;  
...  
default:  
code to be executed if  
n is different  
from all labels;  
}
```

```
$i = 2 ;  
switch ($i) {  
case 0:  
    echo "i equals 0";  
case 1:  
    echo "i equals 1";  
case 2:  
    echo "i equals 2";  
}  
?>
```

❑ PHP switch Statement

```
<?php
switch ($i) {
    case 0:
        echo "i equals 0";
        break;
    case 1:
        echo "i equals 1";
        break;
    case 2:
        echo "i equals 2";
        break;
    default:
        echo "i is not equal to 0, 1 or 2";
}
?>
```

```
<?php
$i = 0;
switch ($i) {
    case 0:
    case 1:
    case 2:
        echo "i is less than 3 ";
        break;
    case 3:
        echo "i is 3";
}
?>
```

❑ PHP switch Statement

❑ ما هي مخرجات البرامج ؟ تتبع

```
<?php
$x = 1;
$y = 3;
switch ($x) {
    case $y - 1:
        print "A";
        break;
    case $y - 2:
        print "B";
        break;
    case $y - 3:
        print "C";
        break;
    case $y - 4:
        print "D";
        break;
}
?>
```



```
<?php
$n='C';
switch($n)
{
    case 'A':
    case 'B':
        echo "A and B\n";
        break;
    case 'C':
    case 'D':
        echo "C and D\n";        break;
    default: echo "A and B\n";
        break;
}
?>
```

❑ PHP Loops

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

- ❑ **while - loops** through a block of code as long as the specified condition is true
- ❑ **do...while - loops** through a block of code once, and then repeats the loop as long as the specified condition is true
- ❑ **for - loops** through a block of code a specified number of times
- ❑ **foreach - loops** through a block of code for each element in an array

□ PHP Loops

while - loops

```
while (condition is true)
{
code to be executed ;
}
```

```
<?php
$x = 10;
while($x <30)
{
$x+=5;
echo "The number is: $x <br>";
}
?>
```

```
The number is: 15
The number is: 20
The number is: 25
The number is: 30
```

❑ PHP Loops

do...while Loop

```
do {  
    code to be executed;  
} while (condition is true);
```

The **do...while loop** will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

```
<?php  
$x = 10;  
  
do {  
    echo "The number is: $x <br>";  
    $x=$x+5;  
} while ($x <= 25);  
?>
```

```
The number is: 15  
The number is: 20  
The number is: 25  
The number is: 30
```


□ PHP Loops

for Loop

The **for loop** is used when you know in advance how many times the script should run.

```
for (init counter; test counter; increment counter)  
{ code to be executed for each iteration; }
```

Parameters:

- **init counter**: Initialize the loop counter value
- **test counter**: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.
- **increment counter**: Increases the loop counter value

```
<?php  
for ($x = 0; $x <= 5; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

```
The number is: 0  
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5
```

❑ PHP Loops

❑ ما هي مخرجات البرامج ؟ تتبع

```
<?php
for ($i = 1; ; $i++)
{
    if ($i > 10) {
        break;
    }
    echo $i;
}
?>
```

```
<?php
    $i = 1;

    for (;;) {
        echo $i++;
        echo "</br>";
    }

?>
```



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٦ و٧)

PHP Arrays

- PHP Array Types
- PHP Array Functions

❑ PHP Arrays

- ❑ An array is a special variable, which can hold more than one value at a time.

PHP Array Types

There are 3 types of array in PHP.

- ❑ Indexed Array
- ❑ Associative Array
- ❑ Multidimensional Array

❑ PHP Arrays Types

❑ PHP Indexed Array

PHP index is represented by number which starts from 0. We can store number, string and object in the PHP array. All PHP array elements are assigned to an index number by default.

There are two ways to define indexed array:

يمكن أن نُهيأ مصفوفة فارغة عبر الدالة `Array()` أو الأقواس `[]` مباشرة

```
$season=array("summer","winter","spring","autumn")
```

```
;
```

```
$season[0]="summer";
```

```
$season[1]="winter";
```

```
$season[2]="spring";
```

```
$season[3]="autumn";
```

□ PHP Arrays Types

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?>
```

دالة حساب عدد عناصر المصفوفة :

`count($ اسم المصفوفة)`

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
for($x = 0; $x < count($cars); $x++)
{
    echo $cars[$x];
    echo "<br>";
}
?>
```

❑ PHP Arrays Types

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo $cars[0] . ", " . $cars[1] . " and " . $cars[2];
?>
```

❑ Loop Through an Indexed Array

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
for($x = 0; $x < 3; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```


❑ PHP Arrays Types

```
<?php
```

```
$arr1 = array("A", "B", "C", "D", "E");
```

```
$arr1[] = "F"; # إضافة عناصر
```

```
$arr1[] = "G";
```

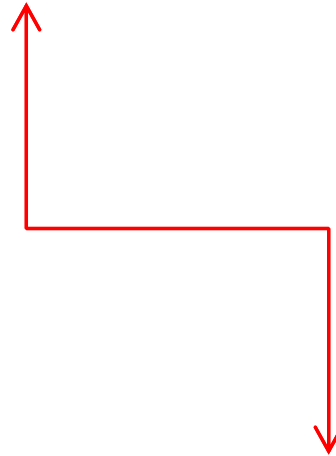
```
for($x=0;$x< count($arr1);$x++)
```

```
{  
  echo $arr1[$x];
```

```
  echo "<br>";
```

```
}
```

```
?>
```



```
$arr1[0] = "22"; # استبدال قيم  
$arr1[3] = "NN";
```

❑ PHP Arrays Types

دالة `print_r()` :

يمكنها طباعة قيمة المتغيرات المعقدة و طباعة محتوى المصفوفة بالكامل وهيكلها ، وعرض المفاتيح والعناصر بتنسيق معين.

```
<?php
```

```
$var1 = "Welcome";
```

```
$var2 = 101;
```

```
$arr = array( "Welcome", "to", "2023");
```

```
print_r($var1);
```

```
echo "<br>";
```

```
print_r($var2);
```

```
echo "<br>";
```

```
print_r($arr);
```

```
?>
```

Outputs:

Welcome

101

Array ([0] => Welcome [1] => to [2] => 2023)

❑ PHP Arrays Types

➤ PHP array_push() Function

```
<?php
$arr = array( "Welcome", "to", "2023");
array_push($arr, 'd', 'f');
echo "<br>";
print_r($arr);
?>
```

array_push(array, var)

✓ يمكن لهذه الدالة إدراج عنصر واحد أو أكثر في نهاية المصفوفة وإرجاع طول المصفوفة الجديدة

➤ PHP array_pop() Function

```
$arr = array( "Welcome", "to", "2023");
array_pop($arr);
echo "<br>";
print_r($arr);
```

array_pop(array)

✓ يمكن لهذه الدالة ازالة عنصر واحد من نهاية المصفوفة

❑ PHP Arrays Types

➤ PHP array_merge () Function

array_merge(array1, array2)

✓ يمكن لهذه الدالة دمج مصفوفتين

```
<?php
$arr1 = array( "Welcome", "to", "2023");
$arr2 = ['d','e','f'];
$new_arr = array_merge($arr1,$arr2);
echo "<br>";
print_r($new_arr);
?>
```

➤ PHP array_sum () Function

array_sum(array)

✓ يمكن لهذه الدالة جمع عناصر مصفوفة

```
$arr = [1,3,5,7,9];
echo array_sum($arr);
```

```
var_dump($arr);
```



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٨)

مفردات المحاضرة :

□ طرق عرض البيانات في **Java Script**

□ الكتابة في عنصر **HTML** ، باستخدام **innerHTML**.

□ الكتابة في مخرجات **HTML** باستخدام **document.write ()**.

□ الكتابة في مخرجات **HTML** باستخدام **window.alert ()**.

□ الكتابة في وحدة تحكم المتصفح ، باستخدام اوامر الـ **console**

□ طباعة صفحة في **Java Script**

□ طرق عرض البيانات في Java Script

- طرق كتابة اكواد JavaScript :
- في مستند HTML يتم إدراج كود JavaScript بين <script> و </script>

١- تكتب في اي مكان في ال body مع مراعاة أسبقية التنفيذ.

```
<body>  
  <script type="text/javascript">  
    //java code  
  </script>  
</body>
```

٢- تكتب ضمن وسم ال head.

```
<head>  
  <script type="text/javascript">  
    //java code  
  </script>  
</head>
```

٣- تكتب في ملف خارجي (عندما يكون حجم التعليمات كبير جداً) ويتم ربطها مع صفحة ال html بحيث نحفظ الملف الخارجي بامتداد js.

```
<script src="java1.js"></script>
```

□ طرق عرض البيانات في Java Script

يمكن لجافا سكريبت "عرض" البيانات بطرق مختلفة :

- الكتابة في عنصر HTML ، باستخدام `innerHTML`.

To access an HTML element, JavaScript can use the `document.getElementById(id)` method.

The `id` attribute defines the HTML element. The `innerHTML` property defines the HTML content:

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = 'ali';
```

```
</script>
```

js1

□ طرق عرض البيانات في Java Script

- الكتابة في مخرجات HTML باستخدام `document.write ()`.

js2

```
<script>  
document.write(5 + 6);  
</script>
```

*سيؤدي استخدام `document.write ()` بعد تحميل مستند HTML إلى حذف كل HTML

الموجود كما موضح في المثال js3

□ طرق عرض البيانات في Java Script

- الكتابة في مخرجات HTML باستخدام `window.alert ()`.

js4

```
<script>  
window.alert(5 + 6);  
</script>
```

```
window.alert(" Computer Sience ");
```

عند تنفيذ هذه الدالة تظهر النتيجة من خلال صندوق حوار عبر المتصفح كما في الشكل :

ⓘ http://localhost/js4.html

Chapter 1 Before th... SQL Tutorial | Essen... C++ Point

localhost says

11

OK

□ طرق عرض البيانات في Java Script

- الكتابة في وحدة تحكم المتصفح ، باستخدام اوامر الـ `console`

تعتبر الطباعة في الـ `console` وسيلة بسيطة للغاية لتحليل و اكتشاف و إصلاح المشاكل الصغيرة في وحدة التحكم .

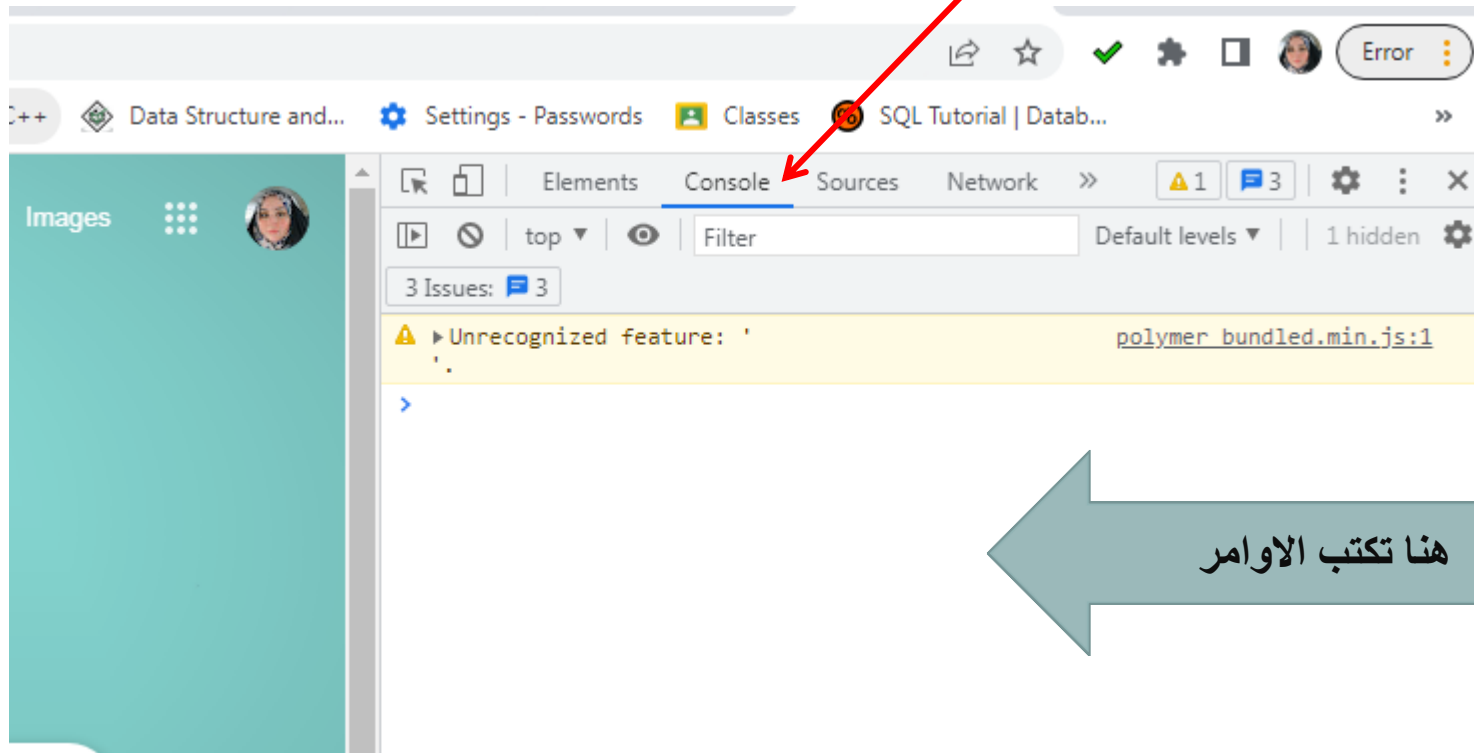
في أغلب المتصفحات المشهورة لفتح وحدة التحكم في المتصفح من خلال النقر على الزر `F12` أو النقر بزر الفأرة الأيمن على أي مكان في المتصفح فيظهر لك خيار اسمه `Inspect Element` فتنقر عليه.

□ فتح الـ `console` داخل المتصفح عن الطریق الضغط على `F12` أو `Ctrl+Shift+K`

□ لتنفيذ الاوامر نضغط على زر " " `Run` أو الضغط على `Ctrl+Enter`.

□ طرق عرض البيانات في Java Script

□ تظهر هذه النافذة نختار التبويب **console**



□ نكتب اوامر الكونسول و البيانات المراد طباعتها داخل المساحة الخاصة به

□ طرق عرض البيانات في Java Script

□ بعض اوامر **console**

□ الامر **console.log** يستخدم لطباعة نص، رقم، مصفوفة، كائن و أي شيء آخر نريده في الكونسول

```
console.log("JavaScript");  
console.log(2022);  
console.log(["HTML", "CSS", "JavaScript"]);
```

```
> console.log("JavaScript");  
console.log(2022);  
console.log(["HTML", "CSS", "JavaScript"]);
```

JavaScript

VM74:1

2022

VM74:2

▶ (3) ['HTML', 'CSS', 'JavaScript']

VM74:3

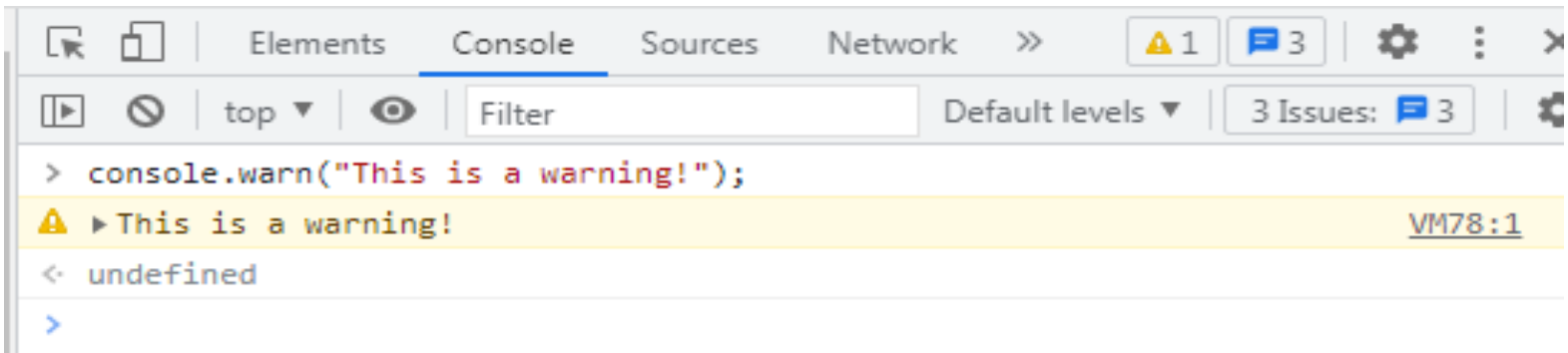
◀ undefined

>

طرق عرض البيانات في Java Script

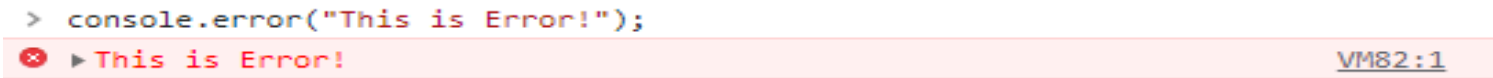
- الأمر `console.warn()`
- يستخدم لطباعة رسائل التحذيرات في الكونسول.

```
console.warn("This is a warning!");
```



- الأمر `console.error()`
- يستخدم لطباعة الأخطاء البرمجية في الكونسول.

```
console.error("This is Error!");
```



□ طرق عرض البيانات في Java Script

في جافا سكربت

نستخدم هذه الأوامر لتجميع ما يتم عرضه في الكونسول في قوائم يمكن فتحها و إغلاقها.

- `console.group()`
 - يستخدم لإنشاء قائمة جديدة مع تحديد عنوانها و محتواها سيكون مفتوح بشكل افتراضي.
 - `console.groupCollapsed()`
 - لإنشاء قائمة جديدة مع تحديد عنوانها و محتواها سيكون مغلق بشكل افتراضي.
 - `console.groupEnd()`
 - يستخدم لتحديد أين تنتهي القائمة.
- ملاحظة:** يمكن وضع القوائم بشكل متداخل.

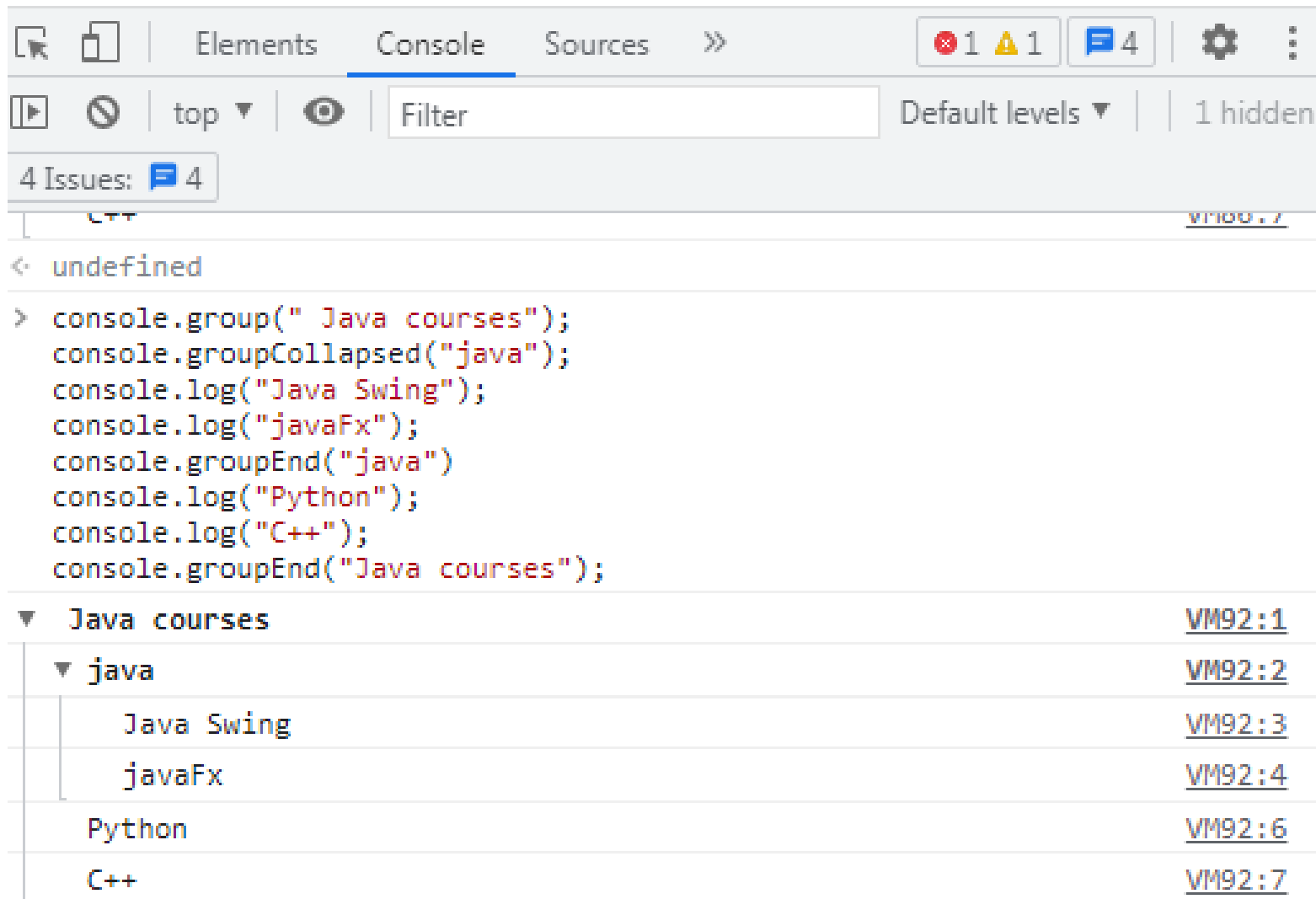
```
console.group(" Java courses");
console.groupCollapsed("java");
console.log("Java Swing");
console.log("javaFx");
console.groupEnd("java")
console.log("Python");
console.log("C++");
console.groupEnd("Java courses");
```

استخدام CSS مع الكونسول في جافاسكربت



```
console.log("%cHarmash", "background:#00418C; color:white; padding:5px;")
```

□ طرق عرض البيانات في Java Script



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The code being executed is:

```
console.group(" Java courses");
console.groupCollapsed("java");
console.log("Java Swing");
console.log("javaFx");
console.groupEnd("java");
console.log("Python");
console.log("C++");
console.groupEnd("Java courses");
```

The output in the console is grouped as follows:

- VM92:1
- < undefined
- > console.group(" Java courses");
- console.groupCollapsed("java");
- console.log("Java Swing");
- console.log("javaFx");
- console.groupEnd("java");
- console.log("Python");
- console.log("C++");
- console.groupEnd("Java courses");

Group Name	Source
Java courses	VM92:1
java	VM92:2
Java Swing	VM92:3
javaFx	VM92:4
Python	VM92:6
C++	VM92:7

□ طباعة صفحة في Java Script

لا تحتوي **JavaScript** على أي كائن طباعة أو طرق طباعة. ولا يمكنها الوصول إلى أجهزة الإخراج الاستثنائية الوحيد هو استدعاء طريقة **window.print ()** في المستعرض لطباعة محتوى الصفحة .

```
<script>  
window.print();  
</script>
```

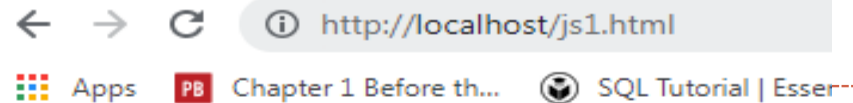
أو

```
<button onclick="window.print()">Print this page</button>
```

□ طرق عرض البيانات في Java Script

js1

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<p id="demi"></p>
<script>
document.getElementById("demo").innerHTML = "Computer systems";
document.getElementById("demi").innerHTML = 7+9;
</script>
</body>
</html>
```



Computer systems

16

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write(5 + 6);
document.write("Computer systems");
</script>
</body>
</html>
```

← → ↻ ⓘ http://localhost/js2.html

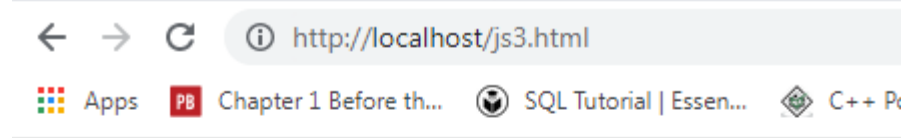
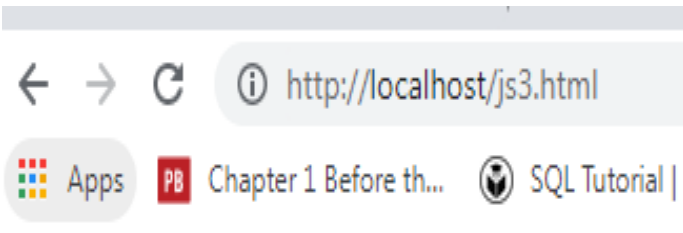
📱 Apps **PB** Chapter 1 Before th... 📄 SQL Tutorial |

11 Computer systems

طرق عرض البيانات في Java Script

js3

```
<!DOCTYPE html>
<html>
<body>
<h1>Welcome to my Java</h1>
<p>My first paragraph.</p>
<button type="button" onclick="document.write(9-5)">Try it</button>
</body>
</html>
```



Welcome to my Java

My first paragraph.

Try it

4



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر



محاضرة (٩)

مفردات المحاضرة :

☐ المتغيرات في Java Script

- ☐ Using **var**
- ☐ Using **let**
- ☐ Using **const**
- ☐ Using **nothing**

- Variables are containers for storing data(storing data values).

4 Ways to Declare a JavaScript Variable:

- Using **var**
- Using **let**
- Using **const**
- Using **nothing**

□ المتغيرات في Java Script

➤ Using var

شروط و قواعد تسمية المتغيرات:

`var [name] = "value";`

- لا تستخدم الأرقام في بداية المتغيرات. هذا غير مسموح به ويسبب خطأ.
- لا تستخدم رموز مثل `@#%^&*-` (في أسماء المتغيرات هذا غير مسموح به ويسبب خطأ).
- الرمزين المسموح بهما هم علامة ال `$` Dollar sign وعلامة `_` underscore.
- يسمح بكتابة `_` underscore سواء في البداية او في منتصف ال `name` لكن ال `_` underscore في البداية يشير الي `constructs`.
- يمكن ان يحتوي اسم المتغير `Variable` علي أحرف وأرقام لا تبدأ اسم المتغير `Variable` بأرقام.
- المتغيرات حساسة لحالة الأحرف - لذا فإن `username` متغير مختلف عن `USERNAME`.
- تحتاج أيضاً إلى تجنب استخدام كلمات `JavaScript` المحجوزة كأسماء متغيرات - بهذا ، نعني الكلمات التي تشكل البنية الفعلية مثل `var` و `function` و `let` و `for` كأسماء متغيرات.

➤ Using var

```
<script>
var username = "Ahmed";
var username1 "Mohamed";
document.write("<b>" + username + "<b><br>");
document.write("<b>" + username1 + "<b><br>");
</script>
```

```
var year = 2018, month = "April", day = 22;
alert(day + "/" + month + "/" + year);
```

```
<p id="demo"></p>
<script>
let carName = "Volvo";
document.getElementById("demo").innerHTML = carName;
</script>
```

➤ Using **let**

- The **let** keyword was introduced in ES6 (2015).
- Variables defined with **let** cannot be Redeclared.
- Variables defined with **let** must be Declared before use.
- Variables defined with **let** have Block Scope.

```
let x, y, z;  
x = 5;  
y = 6;  
z = x + y;
```

➤ Using **let**

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript let varabels </h2>
<p id="demo"></p>
<script>
let x, y, z;
x = 10;
y = 20 ;
z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is " + z + ".";
</script>
</body>
</html>
```

A



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر

محاضرة (١٠)

مفردات المحاضرة :

- ❑ **Types of JavaScript Operators**
 - ❑ **JavaScript Arithmetic Operators**
 - ❑ **JavaScript Assignment Operators**
 - ❑ **JavaScript Logical Operators**



❑ Types of JavaScript Operators

There are different types of JavaScript operators:

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Logical Operators
- Conditional Operators
- Type Operators

❑ JavaScript Arithmetic Operators

JavaScript Arithmetic Operators

Arithmetic Operators are used to perform arithmetic on numbers:

```
let a = 3;  
let x = (100 + 50) * a;
```

```
var x = 15;  
var y = 3;  
var z = x / y;
```

+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

❑ JavaScript Arithmetic Operators

The **modulus** operator (%) returns the division remainder.

```
let x = 5;  
let y = 2;  
let z = x % y;
```

The **decrement** operator (--) decrements numbers.

```
let x = 50;  
x--;
```

The **exponentiation** operator (**) raises the first operand to the power of the second operand.

```
let x = 5;  
let z = x ** 2;
```

Note;

$x ** y$ produces the same result as `Math.pow(x,y)`:

The **increment** operator (++) increments numbers.

```
let x = 8;  
x++; // output 9
```


❑ JavaScript Assignment Operators

➤ Assignment operators assign values to JavaScript variables.

Operator	Example	Same As
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$
**=	$x ** = y$	$x = x ** y$

❑ JavaScript Assignment Operators

Simple Assignment Examples

```
let x = 10;  
let x = 10 + y;
```

The += Operator

Operator adds a value to a variable.

```
let x = 10;  
x += 5;  
document.getElementById("demo").innerHTML = "Value of x is: " + x;
```

The -= Operator

Operator adds a value to a variable.

```
let x = 100;  
x -= 50;  
document.getElementById("demo").innerHTML = "Value of x is: " + x;
```

❑ JavaScript Logical Operators

Operator	Example	Same As
=&&	x &&= y	x = x && (x = y)
=	x = y	x = x (x = y)
=??	x ??= y	x = x ?? (x = y)

The &&= Operator

The Logical AND assignment operator : is used between two values. If the first value is true, the second value is assigned.

```
let x = 10;
```

```
x &&= 5;
```

```
document.write( x ); // output is 5
```

❑ JavaScript Logical Operators

The ||= Operator

The Logical OR assignment operator is used between two values.

If the first value is false, the second value is assigned.

```
let x = undefined;
```

```
x ||= 5;
```

```
document.write( x ); // output is 5
```

The ??= Operator

The Nullish coalescing assignment operator is used between two

values. If the first value is undefined or null, the second value is assigned.

```
let x = null;
```

```
x ??= 5; document.write( x ); // output is 5
```

❑ JavaScript Logical Operators

The ||= Operator

The Logical OR assignment operator is used between two values.

If the first value is false, the second value is assigned.

```
let x = undefined;
```

```
x ||= 5;
```

```
document.write( x ); // output is 5
```

The ??= Operator

The Nullish coalescing assignment operator is used between two

values. If the first value is undefined or null, the second value is assigned.

```
let x = null;
```

```
x ??= 5; document.write( x ); // output is 5
```

❑ JavaScript Logical Operators

What are the outputs of the following code :

```
<script>
let x = 2;
let y = 50;
let c = null;
x ??= y;
c ??= y;
document.write ("Value of x is: " + x) ;
</script>
```

```
document.getElementById("demo").innerHTML = "Value of x is: " + x;
```

<https://www.w3schools.com>

<https://www.javatpoint.com/javascript-tutorial>

<https://www.tutorialspoint.com>



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر

محاضرة (١١)

- ❑ JavaScript Conditional Statements
 - ❑ if statement
 - ❑ if...else statement
 - ❑ if...else if...else statement
 - ❑ JavaScript switch Statement



❑ JavaScript Conditional Statements

In JavaScript, there are three forms of the `if...else` statement.

- ❑ Use `if` to specify a block of code to be executed, if a specified condition is true
- ❑ Use `else` to specify a block of code to be executed, if the same condition is false
- ❑ Use `else if` to specify a new condition to test, if the first condition is false
- ❑ Use `switch` to specify many alternative blocks of code to be executed

❑ JavaScript Conditional Statements

The **if** Statement

Use the **if** statement to specify a block of JavaScript code to be executed if a condition is true.

```
if (condition) {  
    // block of code to be executed if the condition is true  
}
```

```
<script>  
var n=8;  
if(n%2==0){  
document.write("n is even number");  
}  
</script>
```

❑ JavaScript Conditional Statements

The else Statement

Use the **else** statement to specify a block of code to be executed if the condition is false.

```
if (condition) {  
    // block of code to be executed if the condition is true  
} else {  
    // block of code to be executed if the condition is false  
}
```

```
<script>  
var n=8;  
if(n%2==0){  
document.write("n is even number");  
} else{  
document.write("a is odd number");  
}  
</script>
```

❑ JavaScript Conditional Statements

The else if Statement

Use the **else if** statement to specify a new condition if the first condition is false.

```
if (condition1) {  
    // block of code to be  
    executed if condition1 is  
    true  
} else if (condition2) {  
    // block of code to be  
    executed if the condition1  
    is false and condition2 is  
    true  
} else {  
    // block of code to be  
    executed if the condition1  
    is false and condition2 is  
    false  
}
```

```
<script>  
var a=20;  
if(a==10){  
document.write("a is equal to 10"); }  
else if(a==15){  
document.write("a is equal to 15");  
}  
else if(a==20){  
document.write("a is equal to 20");  
}  
else{  
document.write("a is not equal to 10,  
15 or 20");  
}  
</script>
```

❑ JavaScript Conditional Statements

The JavaScript Switch Statement

Use the **switch** statement to select one of many code blocks to be executed.

```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}
```

This is how it works:

- ❑ The switch expression is evaluated once.
- ❑ The value of the expression is compared with the values of each case.
- ❑ If there is a match, the associated block of code is executed.
- ❑ If there is no match, the default code block is executed.

```
<p id="demo"></p>
```

```
<script>
```

```
let day;
```

```
switch (new Date().getDay()) {
```

```
  case 0:
```

```
    day = "Sunday";
```

```
    break;
```

```
  case 1:
```

```
    day = "Monday";
```

```
    break;
```

```
  case 2:
```

```
    day = "Tuesday";
```

```
    break;
```

```
  case 3:
```

```
    day = "Wednesday";
```

```
    break;
```

```
  case 4:
```

```
    day = "Thursday";
```

```
    break;
```

```
  case 5:
```

```
    day = "Friday";
```

```
    break;
```

```
  case 6:
```

```
    day = "Saturday";
```

```
}
```

```
document.getElementById("demo").innerHTML = "Today is " + day;
```

```
</script>
```

Example

The `getDay()` method returns the weekday as a number between 0 and 6 Sunday=0 ,Monday=1 , Tuesday=2

- This example uses the week day number to calculate the weekday name:

❑ JavaScript Conditional Statements

The **default** Keyword

The default keyword specifies the code to run if there is no case match:

Example

If today is neither Saturday (6) nor Sunday (0), write a default message:

```
switch (new Date().getDay()) {  
  case 6:  
    text = "Today is Saturday";  
    break;  
  case 0:  
    text = "Today is Sunday";  
    break;  
  default:  
    text = " Have a nice day ";  
} document.getElementById("demo").innerHTML = text;
```


❑ JavaScript Conditional Statements

The **default** Keyword

The default keyword specifies the code to run if there is no case match:

The default case does not have to be the last case in a switch block.

Example

If today is neither Saturday (6) nor Sunday (0), write a default message:

```
switch (new Date().getDay()) {
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = " Have a nice day ";
} document.getElementById("demo").innerHTML = text;
```

❑ JavaScript Conditional Statements

Strict Comparison

Switch cases use strict comparison (===).

The values must be of the same type to match.

لاحظ الفرق في نتيجة كل من ما يأتي:

```
let x = "0";
switch (x) {
  case 0:
    text = "Off";
    break;
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
}
```

```
let x = "0";
switch (x) {
  case "0":
    text = "Off";
    break;
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
}
```

<https://www.w3schools.com>

<https://www.javatpoint.com/javascript-tutorial>

<https://www.tutorialspoint.com>



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر

محاضرة (١٢)

مفردات المحاضرة:

- ❑ **Arrays in JavaScript**
 - ❑ **Create Arrays in JavaScript**
 - ❑ **Print Arrays in JavaScript**
 - ❑ **Add & removes elements in Array**
 - ❑ **Accessing Array Elements**



❑ Arrays in JavaScript

- ❑ An array is a special variable, which can hold more than one value:

```
const cars = ["Saab", "Volvo", "BMW"];
```

❑ Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

```
const array_name = [item1, item2, ...];
```

It is a common practice to declare arrays with the `const` keyword.

Spaces and line breaks are not important. A declaration can span multiple lines

❑ Arrays in JavaScript

❑ يوجد طريقتان لإنشاء المصفوفة في جافا سكريبت:
1. التعريف باستخدام الأقواس [].

```
const cars = [];  
cars[0] = "Saab";  
cars[1] = "Volvo";  
cars[2] = "BMW";
```

```
var cars = ["Saab", "Volvo", "BMW"] ;
```

2- التعريف بواسطة الـ **constructor** باستخدام كلمة **new**.

```
var cars = new Array("Saab", "Volvo", "BMW");
```

❑ للوصول الى عنصر معين في المصفوفة يتم ذلك عن طريق **رقم الفهرس index** للعنصر.
كما في المثال الاتي :

```
var name = cars[0];
```

□ Arrays in JavaScript

□ **تبدیل عنصر معین فی مصفوفة یكون بذكر فهرس العنصر و القيمة :**

```
var cars = ["Saab", "Volvo", "BMW"] ;  
cars = [٢]= "333" ;
```

□ **طباعة المصفوفة باكملها ، يمكن استخدام دوال الاخراج في javascript :**

```
const cars = ["Saab", "Volvo", "BMW"];  
document.getElementById("demo").innerHTML = cars;
```

□ **طباعة عنصر معین او عناصر معينة من مصفوفة یكون بذكر فهرس العنصر:**

```
const cars = ["Saab", "Volvo", "BMW"];  
document.write(cars[2] );
```


❑ JavaScript Arithmetic Operators

❑ إضافة العناصر للمصفوفات

❑ تستخدم دوال `push` و `unshift` لإضافة عنصر إلى المصفوفة:

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Kiwi");
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.unshift("Lemon");
```

❑ ما الفرق بين `push` و `unshift` ?

❑ إزالة عناصر من مصفوفة

❑ تستخدم دوال `pop` و `shift` لإضافة عنصر إلى المصفوفة : **واجب**

❑ Arrays in JavaScript

❑ دمج مصفوفات

❑ تستخدم دالة لدمج مصفوفتين او اكثر و تكوين مصفوفة جديدة :

```
const arr1 = ["L", "K"];  
const arr2 = ["E", "T", "L"];  
const myarr = arr1.concat(arr2); // دمج مصفوفتين
```

```
const arr1 = ["L", "K"];  
const arr2 = ["E", "T", "L"];  
const arr3 = ["c", "d", "A"];  
const myarr = arr1.concat(arr2, arr3); // دمج اكثر من مصفوفة
```

❑ يمكن دمج عنصر (قيمة) مع مصفوفة :

```
const arr1 = ["E", "T", "L"];  
const arr2 = arr1.concat("P");
```

❑ Arrays in JavaScript

❑ خاصية **length** ترجع طول المصفوفة (عدد عناصر المصفوفة) :

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
document.write( fruits.length);
```

You can have objects in an Array. functions in an Array. arrays in an Array:

```
myArray[0] = new Date();  
myArray[1] = myFunction;  
myArray[2] = myCars;
```

❑ في javascript يمكن لعناصر المصفوفة ان تكون مختلفة في النوع البياني و ان تحتوي على objects مختلفة .

<https://www.w3schools.com>

<https://www.javatpoint.com/javascript-tutorial>

<https://www.tutorialspoint.com>



**Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques**



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر

محاضرة (١٣)

مفردات المحاضرة:

Loops in JavaScript

- for Loop
- while - loops
- do...while Loop



❑ Javascript Loops

JavaScript supports different kinds of loops:

- **for** - loops through a block of code a number of times
- **for/in** - loops through the properties of an object
- **for/of** - loops through the values of an iterable object
- **while** - loops through a block of code while a specified condition is true
- **do/while** - also loops through a block of code while a specified condition is true

❑ Javascript Loops

The **for statement** in JavaScript has the same syntax as in Java and C. It has three parts:

- **Initialization** - Initializes the iterator variable *i*. In this example, we initialize *i* to 0.
- **Condition** - As long as the condition is met, the loop continues to execute. In this example, we check that *i* is less than 3.
- **Increment** - A directive which increments the iterator. In our case, we increment it by 1 on every loop.

We can also write a shorter notation for the statement by inserting the variable definition inside the for loop and incrementing using the ++ operator.

```
for (Initialization; Condition; iteration) {  
    // code block to be executed  
}
```


❑ Javascript Loops

```
<script>
let text = "";
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
```

EX1

```
<p id="demo"></p>
<script>
const cars = ["BMW", "Volvo", "Saab", "Ford"];
let i, text;
for (i = 0, text = ""; i < cars.length; i++) {
  text += cars[i] + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
```

EX2

□ Javascript Loops

for loop with html table

□ عرض مجموعة بيانات محفوظة بداخل مصفوفة array عن طريق حلقات التكرار for loop بداخل جدول html عن طريق document.write في لغة جافا سكريبت.

```
var courses = ['HTML', 'CSS', 'PHP', 'JS', 'jQuery', 'MY SQL', 'Angular'];
document.write("<table class='table'>");
for (var i=0; i < courses.length; i++) { document.write("<tr>");
document.write("<td>" + (i + 1) + "</td><td>" + courses[i] + "</td>");
document.write("</tr>"); }
document.write("</table>");
```

1	HTML
2	CSS
3	PHP
4	JS
5	jQuery
6	MY SQL
7	Angular

□ Javascript Loops

```
let i = 0;  
while(expression)  
{ الكودات الواجب تنفيذها }
```

- condition هو متغير أو شرط وبناء عليه اذا كان صحيحاً true يتم تنفيذ حلقات التكرار.
- يمكن أن يتم تعريف متغير variable خارج حلقة التكرار while loop والتحقق من أنه كان صحيحاً ام غير صحيح. true or false.
- يمكن استخدام عوامل المقارنة [Comparison Operators](#) للتحقق من قيمة ما مقابل قيمة اخرى.
- لا يعمل التكرار إلا في حالة واحدة إذا كان قيمة condition صحيحة. true.
- يتم تجاهل الأوامر البرمجية المحجوزة بداخل الكُتلة البرمجية بين الأقواس curly braces في حال كان الشرط غير صحيح. false.
- قيمة condition تكون true أو false وبناء عليه يعمل التكرار مره أو عدة مرات في

❑ Javascript Loops

while - loops

EX1

```
let i = 0;
while(i < 10)
{ i=i+2;
document.write("The number is " + i + "<br>") ;
}
```

The number is 2
The number is 4
The number is 6
The number is 8
The number is 10

EX2

```
let i = 0;
while(i < 10)
{
document.write("The number is " + i + "<br>") ;
i=i+2;}
}
```

The number is 0
The number is 2
The number is 4
The number is 6
The number is 8

○ لاحظ الفرق في مخرجات EX1,EX2

○ امثلة:

```
<html>
<body>
<h1>JavaScript For Loop</h1>
<script>
for (i = 1; i <= 10; i++) {
    document.write("Text " + i + "<br>");
}
</script>
</body>
</html>
```

```
<html>
<body>
<script>
let courses= ["HTML", "CSS", "JavaScript", "CSS"];
for (let i = 0;i < courses.length;i++){
    document.write((i + 1) + "- " + courses[i] + "<br>");
}
</script>
</body>
</html>
```

<https://www.w3schools.com>

<https://www.javatpoint.com/javascript-tutorial>

<https://www.tutorialspoint.com>



Southern Technical University
Qurna Technical Institute
Department of
Computer Systems Techniques



المرحلة : الثانية

المادة : تصميم المواقع الالكترونية المتقدم

مدرس المادة : م. اسراء محمود حيدر

محاضرة (15)

مفردات المحاضرة :

- Insert Data Into MySQL
- Delet Data Into MySQL
- Updatet Data Into MySQL



❑ Insert Data Into MySQL

After a database and a table have been created, we can start adding data in them.

Here are some syntax rules to follow:

- The SQL query must be quoted in PHP .
- String values inside the SQL query must be quoted.
- Numeric values must not be quoted .
- The word NULL must not be quoted .
- The INSERT INTO statement is used to add new records to a MySQL table:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

❑ Insert Data Into MySQL

المثال التالي يوضح عملية اضافة قيد الى الجدول **Info** و الذي تم انشاؤه ضمن قاعدة بيانات **student**

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "student";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);}
$sql = "INSERT INTO Info (firstname, lastname, email)
VALUES ('Ali', 'Ahmed', 'Ali@gmail.com')";
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";} else {
    echo "Error: " . $sql . "<br>" . $conn->error;}
$conn->close();
```

?>

❑ Delete Data From a MySQL Table

The DELETE statement is used to delete records from a table:

```
DELETE FROM table_name  
WHERE some_column = some_value
```

مثال يوضح عملية حذف القيد ID 3 من الجدول **Info** و الذي تم انشاؤه ضمن قاعدة بيانات **student**

id	firstname	lastname	age
1	Ali	M.	18
2	Mohammed	A.	22
3	Rana	D.	23

❑ Delete Data From a MySQL Table

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = " Student";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);}
// sql to delete a record
$sql = "DELETE FROM Info WHERE id=3";
if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;}
$conn->close();
?>
```

□ Update Data in a MySQL Table

The UPDATE statement is used to update existing records in a table:

```
UPDATE table_name
```

```
SET column1=value, column2=value2,...
```

```
WHERE some_column=some_value
```

مثال يوضح عملية تحديث بيانات القيد 2 ID من الجدول **Info** و الذي تم انشاؤه ضمن قاعدة بيانات **student**

id	firstname	lastname	age
1	Ali	M.	18
2	Mohammed	A.	22
3	Rana	D.	23

❑ Update in MySQL Table

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "Student";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);}
$sql = "UPDATE Info SET lastname='MM' WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;}
$conn->close();
?>
```

<https://www.w3schools.com>

<https://www.javatpoint.com/javascript-tutorial>

<https://www.tutorialspoint.com>